*PROJECT REPORT*

**PROGRAM LOGIC**

This project uses Clojure via InelliJ IDE with the Cursive add-on. This program will sort a large list of integers from the provided .txt file which contains one million integer values. The program uses the merge sort algorithm and sorts from smallest to largest. The program will then run the same algorithm with 2, 4, 8, 16, 32, and 64 threads. It will measure the time needed to sort the .txt file using parallelism and the above number of threads. The purpose of this report is to explain the findings of this code.

***CODE DESCRIPTION***

***TESTING RESULTS AND CONCLUSIONS***

This code has been run 3 times and the average of each result is provided below:

|  |  |
| --- | --- |
| Average run time for 100lines.txt file | |
| Number of Threads | Average Run Time |
| 1 |  |
| 2 |  |
| 4 |  |
| 8 |  |
| 16 |  |
| 32 |  |
| 64 |  |

|  |  |
| --- | --- |
| Average run time for 3000lines.txt file | |
| Number of Threads | Average Run Time |
| 1 |  |
| 2 |  |
| 4 |  |
| 8 |  |
| 16 |  |
| 32 |  |
| 64 |  |

|  |  |
| --- | --- |
| Average run time for numbers.txt file | |
| Number of Threads | Average Run Time |
| 1 |  |
| 2 |  |
| 4 |  |
| 8 |  |
| 16 |  |
| 32 |  |
| 64 |  |

As to be expected, the numbers.txt file took the most amount of time to run through due to the fact that the numbers.txt file contains the most items to sort through.

***REFERENCES***

*Resources Used*

Stuart Sierra, S. H. (n.d.). *API for clojure.string - Clojure v1.10.2 (stable)*. Retrieved from Clojure: https://clojure.github.io/clojure/clojure.string-api.html

https://clojuredocs.org/clojure.core

https://clojuredocs.org/clojure.core/pmap

Retrieved from time https://clojuredocs.org/clojure.core/time

https://stackoverflow.com/questions/55770730/trying-to-read-a-text-file-in-clojure-and-insert-the-data-into-a-list-or-a-vecto

*Code References*

https://gist.github.com/baabelfish/6573984